

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-267877

(43) 公開日 平成 8 年 (1996) 10 月 15 日



(51) Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
B 4 1 J 29/38			B 4 1 J 29/38	Z
G 0 6 F 3/12			G 0 6 F 3/12	A

審査請求 未請求 請求項の数 8 O L (全 15 頁)

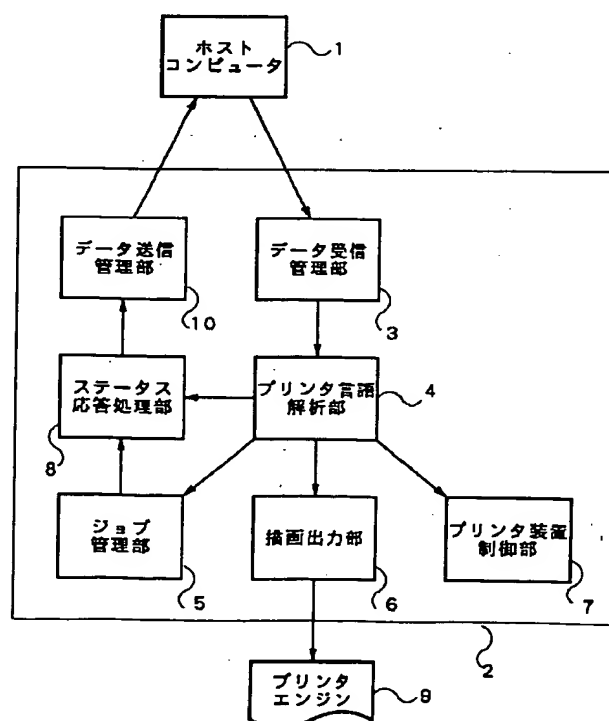
(21) 出願番号	特願平7-69003	(71) 出願人	000001007 キヤノン株式会社 東京都大田区下丸子3丁目30番2号
(22) 出願日	平成7年(1995)3月28日	(72) 発明者	的場 暢 東京都大田区下丸子3丁目30番2号キヤノン株式会社内
		(74) 代理人	弁理士 丸島 儀一

(54) 【発明の名称】 プリンタ装置及びその制御方法

(57) 【要約】

【目的】 ホストコンピュータ側で確実にその状態を正確に把握することができるプリンタ装置及びその制御方法を提供することを目的とする。

【構成】 ホストコンピュータ1から受信したジョブ単位の水タそれぞれに固有の識別子を割り当て、該割り当てられた識別子をデータ元に対して返信すると共に受信したジョブ単位の水タに対する処理の完了応答である情報をデータ元に返信する。



【特許請求の範囲】

【請求項1】 データ元からのジョブ単位の前記データに従ってプリント処理を行うプリンタ装置であって、前記ジョブ単位の前記データを受信する受信手段と、該受信手段により受信したジョブ単位の前記データそれぞれに固有の識別子を割り当てる割り当て手段と、前記受信手段により受信したジョブ単位の前記データに対する処理の完了応答である情報を、前記割り当て手段で当該ジョブ単位の前記データに対して割り当てられた識別子と共にデータ元に返信する情報返信手段と、を備えることを特徴とするプリンタ装置。

【請求項2】 前記情報返信手段は、受信したジョブ単位の前記データの種類の判定し、ジョブ単位の前記データの種類の応じて、プリンタ装置の状態と共に識別子を返信することを特徴とする請求項1記載のプリンタ装置。

【請求項3】 データ元からのジョブ単位の前記データに従ってプリント処理を行うプリンタ装置の制御方法であって、前記ジョブ単位の前記データを受信する受信工程と、該受信工程により受信したジョブ単位の前記データそれぞれに固有の識別子を割り当てる割り当て工程と、前記受信工程により受信したジョブ単位の前記データに対する処理の完了応答である情報を、前記割り当て手段で当該ジョブ単位の前記データに対して割り当てられた識別子と共にデータ元に返信する情報返信工程と、を備えることを特徴とするプリンタ装置の制御方法。

【請求項4】 前記情報返信工程は、受信したジョブ単位の前記データの種類の判定し、ジョブ単位の前記データの種類の応じて、プリンタ装置の状態と共に識別子を返信することを特徴とする請求項3記載のプリンタ装置の制御方法。

【請求項5】 データ元からのジョブ単位の前記データに従ってプリント処理を行うプリンタ装置であって、前記ジョブ単位の前記データ及び当該ジョブ単位の前記データ毎に割り当てられた固有の識別子を受信する受信手段と、前記受信手段により受信したジョブ単位の前記データに対する処理の完了応答である情報を、当該ジョブ単位の前記データに対して割り当てられた識別子と共にデータ元に返信する情報返信手段と、を備えることを特徴とするプリンタ装置。

【請求項6】 前記情報返信手段は、受信したジョブ単位の前記データの種類の判定し、ジョブ単位の前記データの種類の応じて、プリンタ装置の状態と共に識別子を返信することを特徴とする請求項5記載のプリンタ装置。

【請求項7】 データ元からのジョブ単位の前記データに従ってプリント処理を行うプリンタ装置の制御方法であって、前記ジョブ単位の前記データ及び当該ジョブ単位の前記データ毎に割り当てられた固有の識別子を受信する受信工程と、前記受信工程により受信したジョブ単位の前記データに対する

処理の完了応答である情報を、当該ジョブ単位の前記データに対して割り当てられた識別子と共にデータ元に返信する情報返信工程と、を備えることを特徴とするプリンタ装置の制御方法。

【請求項8】 前記情報返信工程は、受信したジョブ単位の前記データの種類の判定し、ジョブ単位の前記データの種類の応じて、プリンタ装置の状態と共に識別子を返信することを特徴とする請求項7記載のプリンタ装置の制御方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、例えばホストコンピュータから受け取ったデータを解析し、出力処理を行なうプリンタ装置及びその制御方法に関するものである。

【0002】

【従来の技術】 従来のプリンタ装置として、ホストコンピュータから送られてきたコマンドに従って、文字や図形やイメージなどを描画し、文書出力を行なったり、プリンタの初期化などの装置の制御を行なったりするものが知られている。

【0003】 また同様に、ホストから送られてきたコマンドに従って、プリンタ装置の状態や、プリンタ環境の設定、文書出力の経過、使用するフォントなどの情報などを、双方向インターフェースを介して、ホストへ応答する機能を持ったプリンタ装置も知られている。

【0004】 このような各種プリンタ情報を、コマンドに従いホストに伝達する機能を「ステータス応答」と呼ぶ。

【0005】 ステータス応答で用いられる双方向インターフェースとしては、従来から一般的なRS-232Cや、近年登場したBi-Centronics、イーサネットなど色々な方式がある。

【0006】 近年のプリンタ装置では、従来の単方向インターフェースに代わり、これら双方向インターフェースが標準となりつつある。今後この双方向インターフェースを用いて、ステータス応答を利用するプリンタシステムがますます増加していくと考えられる。

【0007】 ステータス応答の機能を用いることで、ホストコンピュータは、接続されているプリンタの状態を把握することができる。

【0008】 例えば、プリンタ装置が現在印字可能な状態であるかをステータス応答させたり、紙切れや紙詰まりなどの障害が発生した場合にそれを連絡してもらったりすれば、ホストコンピュータから離れたところにプリンタ装置が設置されていたりしても、いちいちプリンタ装置の状態を確認しに向いたりする必要がなくなる。

【0009】 また、そのプリンタ装置で使用できるフォントの種類、描画できる図形の種類、実行できるプリンタ言語の種類などの情報をステータス応答してもらえば、文書出力データを作成する時に、どのようなフォン

トを使い、どのプリンタ言語を使って描画すればいいか、などの指針となる。

【0010】従来の単方向インターフェースプリンタ装置では、ホストコンピュータ側は、データを送ったら送ったっきりで、プリンタ装置が正しく描画してくれたか、何か障害が起こって停止しているのではないかなどといったことが全く把握できず、また、データ作成時にも、このプリンタ装置には、このフォントやこの描画機能が用意されているはずだから、といった推測で描画データを作らなければならなかった。

【0011】今後は、この双方向インターフェースとステータス応答機能を用いて、ホスト側でプリンタ装置の情報を獲得して管理するようなシステムが一般的になってくると考えられる。

【0012】双方向インターフェースを用いてステータス応答を行なう場合、まずステータス応答を要求するコマンドをホストがプリンタに対して発行する。

【0013】プリンタ側では、これを言語解析し、ステータス応答を要求しているコマンドであると認識し、該当するプリンタの情報を調べ、これを転送データ形式で記述し、ホストコンピュータへ送信する。

【0014】ホストではこれを受け取り、データ解析して、プリンタの状態を知ることになる。

【0015】これでわかるように、ホストとプリンタのインターフェースとしては、ホストからプリンタへの送信と、プリンタからホストへの返信の、2つの方向がある。

【0016】この2方向のデータのやりとりは、その双方向通信の形式により異なる。

【0017】例えば、送信と返信が完全に独立したタイミングで行なわれる方式もあれば、送信か返信のどちらか片方だけが一時期に実行でき、必要に応じて送信、返信を切替える方式もある。

【0018】プリンタ装置の場合、通常の使われ方ではホストからプリンタ装置へのデータの流れが、そのほとんどを占める。

【0019】このため、プリンタからホストへの転送よりも、ホストからプリンタへの転送に重点をおき、この方向の流れが効率良く実行されるように、そのインターフェースのアルゴリズムや、ハードウェアなどが調整されている。

【0020】このため、送信と返信が、必ずしも1対1のタイミングで実行されるとは限らない。

【0021】例えば、ステータス応答を含む出力データを送った場合、出力処理を優先するために、ある程度出力データ処理をしてからでないと、ステータス応答を返せない場合がある。応答をするためには、出力データの受け取りを一時ストップしたり、CPUの処理をそれに割り振らなければならず、効率良いプリンタ処理を妨げてしまうからである。

【0022】これはホスト側の送受信部の動作についても同様である。

【0023】一般的に、ホスト側がプリンタからの応答を受ける処理は、出力データを送る処理よりも、プライオリティを低くしている。

【0024】また、転送速度もプリンタへの送信速度に比べて、プリンタからの受信はスピードが遅い場合が多い。

【0025】このように一般のプリンタ装置では、ホストからプリンタへのデータ転送に重点がおかれており、応答に関しては扱いが低いために、ステータス応答要求コマンドを発行しても、すぐに応答が戻ってくるとは限らないのである。

【0026】また、ホスト側、データ送受信を行なうインターフェース、プリンタ側のそれぞれの処理速度や、処理のタイミングの違いによるズレを抑えるために、それぞれの接点において、データバッファを設けることが一般的になっている。

【0027】例えば、ホストからインターフェースを通して、プリンタ装置にデータを送る時には、プリンタ装置の入口に受信バッファと呼ばれるデータバッファを設けておく。インターフェースから送られてきたデータは、一旦このバッファに蓄積される。この蓄積処理は、インターフェースの処理速度、処理タイミングに合わせて行なわれるので、インターフェースにとっての負担を軽減し、そのインターフェースの最適な条件で処理を行なうことが出来る。

【0028】またプリンタ装置側も、プリンタ装置の処理速度、処理タイミングで、その受信バッファからデータを受けることが出来、同様に最適な条件で処理が行なえる訳である。

【0029】しかし、一旦データを溜めておくということは、ホストがデータを発行してそれがすぐに処理されるのではなく、ある程度の時間を置いた上で処理されるということを意味する。

【0030】今ホストがステータス応答要求を指令したとしても、その指令データは、一旦受信バッファなどに蓄えられ、プリンタ側がそのバッファを順番に読み込み、その指令データを受け取った時点で始めて処理が開始される訳である。

【0031】このため、データ搬送経路中にこのようなバッファがある場合には、ステータスの要求発行をしてから、応答の返信が届くまでには、ある程度時間がかかってしまう。

【0032】次に、プリンタ装置で行なわれるステータス応答処理である。

【0033】応答の内容によっては、その情報を獲得するのに時間がかかるものがある。例えば、内蔵されているフォントの全文字をサーチして情報を得るようなステータス応答を指令したとする。用意されているフォント

の状態にもよるが、場合によっては数秒を要することがある。

【0034】また、プリンタの状態が変化したら、それをホスト側に連絡するようなステータス応答を要求したとする。

【0035】例えば、文書出力を行なっていて、全ページが完全に出力し終わったら、ホスト側にそれを通知するようなステータス応答を要求する。

【0036】すると、要求コマンドを発行してから、全ページ排出し終わって、応答が返るまで数十秒から数分の時間差が生じることがある。

【0037】このような時に、応答が返るまでホストからの出力データ送信を止める訳にはいかないので、要求コマンドと、ステータス応答のタイミングは、完全にずれることとなる。

【0038】以上に述べたように、ステータス応答の返信は、必ずしもホストがステータス応答要求コマンドを発行してすぐに返されるとは限らない。

【0039】即時に応答される場合もあれば、数秒から数分以上も経って応答される場合もある。ホストコンピュータ側では、これを受け、解析しプリンタの状態を知ることとなる。

【0040】従来のプリンタ装置のようにステータス応答を用いる頻度が低い場合には、多少の時間差があっても、許容範囲として処理することができた。

【0041】しかし、今後ますますこのステータス応答を用いたプリンタ装置との情報交換は盛んになり、その使用頻度も高くなっていく。

【0042】こうなると、この送信と応答のタイミングのズレは、大きな問題をはらんでくる。

【0043】さて、具体的に従来のプリンタ装置における問題点を説明する前に、実際の文書出力について説明しておこう。

【0044】一般に一つのドキュメントをプリントする場合に、ホストコンピュータなどからプリンタ装置に送られる出力データは、図7に示すような形態を持っている。この出力データのひとかたまりを「ジョブ」と呼ぶ。

【0045】まず出力データのジョブ開始を宣言する(1)。これはプリンタ装置に対して出力データが送られて来たことを知らせる働きをする。

【0046】次にプリンタ環境を初期化する(2)。プリンタの環境は、それ以前に送られてきたデータによりいろいろな状態に変化している可能性がある。このまま新しいデータを処理すると、その変化した環境の影響を受けてしまい、予定した通りにプリントできない場合がある。そこで出力処理を開始する前にプリンタの環境を初期化することで、このそれ以前の影響を排除する。

【0047】さらにこれから始める出力で用いるプリンタ環境を設定する(3)。例えばプリントする用紙のサ

イズを設定したり、印字で用いるフォントをホストコンピュータからダウンロードして登録しておいたりする。この前準備により、これに続くプリント処理が正しく実行される。

【0048】この(1)～(3)の一連の処理をジョブの前処理と呼ぶ。

【0049】そして実際の出力を行なう(4)。出力は文字や図形やイメージなどを描画するデータを記述し、1ページ分のデータ記述が終わったら、排出命令を発行するのが一般的である。これを出力するページ分繰り返すことで、複数ページに渡るドキュメントがプリントされる。

【0050】全ページのプリント指示が終わったら、後処理として使用したプリンタの環境を元に戻す(5)。変更したプリンタ環境を元に戻したり、使用した登録フォントなどを削除したりして、次に送られてくるであろう別の出力のための環境を整えておく。

【0051】最後に出力データのジョブ終了を宣言する(6)。これによりプリンタ装置は、一つのデータ処理が完了したことを認識する。

【0052】この(5)～(6)の一連の処理をジョブの後処理と呼ぶ。

【0053】以上は一般的な例であり、実際にはもう少し複雑でこのように明確に分けられない場合もある。例えば、(1)の開始宣言と同時に自動的に(2)の初期化がされてしまったり、(5)の後処理が省略されたりすることもある。しかし、概ねこのような形態になっていると考えて差し支えない。

【0054】この(1)～(6)の一連のデータのかたまりを、一般的に「ジョブ」(または「プリントジョブ」など)と呼ぶ。ホストコンピュータから一つのドキュメントを出力する時には通常の場合、プリンタ装置はこのジョブの単位でデータを処理していく。

【0055】ホストから複数のドキュメントを出力する場合には、順番にこのドキュメント出力に該当する、複数のジョブを送ることで実現する。プリンタ装置は送られてきた複数のジョブを一つ一つのジョブの単位で順番に処理しプリントする。

【0056】またネットワークなどのように複数のコンピュータが接続されていて、複数のユーザがそれぞれのドキュメント出力を行なうような場合にも、それぞれから送られてきたジョブを、一つ一つのジョブの単位で順番に処理してプリントする。

【0057】このように複数のホスト、複数のユーザ、複数のドキュメントの出力を、このようなプリントシステムでは、ジョブの単位で処理することで、混乱無く対処している訳である。

【0058】それでは次に、以上のようなジョブの各段階において、ステータス応答が実際にどのように使われているのかを説明しよう。

【0059】例えばまず、図7のプリンタ環境設定(3)の段階だが、プリンタ環境を設定するためには、そのプリンタにどのようなプリンタ環境項目が存在しているか、その項目に対してどのような設定値が選択できるのか、などといった情報が判っていないと行なえない。

【0060】例えば大きい用紙に印字したい時、そのプリンタで利用できる最大の用紙サイズがA3版なのかB4版なのかによって、記述されるデータは大きくことなる。これらはプリンタの機種やバージョンによって異なっている可能性がある。B4版の大きさまでしか対応していない機種なのかもしれない。或いは、対応している機種でも、その時に装着されている紙カセットなどの装着部品によっては、A3版での印字が出来ないかもしれない。

【0061】またプリンタ側に用意されているフォントの種類なども、ドキュメントを記述するためには重要な情報である。例えば、72ポイントの大きさの文字を印字したい時、そのフォントがプリンタに無いと正しく印字できなくなる。場合によっては必要なフォントをホスト側からダウンロードしてプリンタ内に登録しておかなければならない。

【0062】イメージデータのような大量データを送る時に、もしプリンタ側に圧縮されたデータを受けとる能力が備わっているならば、データを圧縮してから送ることで、データ転送時間を大幅に削減することが可能となる。このような便利な機能が備わっていても、それを知らなければ、ホストは無駄に処理に時間を取られてしまう。

【0063】このようにプリンタ側の状態により、出力データの記述内容などが変化してしまう場合が多々発生するので、前もってプリンタの状態を調査しておく必要がある。このためにステータス応答を用いる。ステータス応答では、使用できる用紙の大きさやフォントの種類、空いているRAMの容量などといった、プリンタ装置のあらゆる情報を得ることが出来る。プリンタの状態や能力を知ること、出力データを最適な形態で記述することが可能となり、そのプリンタの性能を十分に引き出すことができる。

【0064】次に例えば、実際の各ページ出力(4)の段階では、刻々と変化するプリンタの状況を、ステータス応答によりホストに伝達する。

【0065】例えば、出力している途中で用紙が無くなってしまった場合には、紙切れをあらわすステータス応答「STATUS PAPER-OUT」などをホストに通知する。これを受けてホストでは、ユーザに対して紙切れの警告をコンピュータ端末の画面などに表示する。さらに何らかの障害が発生して、正しくプリントが行なえなかった場合には、再度プリントをやり直すために、データの再送要求のステータス応答を送る。また順調にプリントされて

いる場合にも、現在何ページまで印字できたかなどといった、途中状況の通知を行なう。

【0066】このようにプリンタの環境や能力、刻々と変化する状況等の情報を、ステータス応答により伝達することで、ホストコンピュータ側でプリンタの状態を把握することが可能となる訳である。

【0067】

【発明が解決しようとする課題】それではこれらジョブの概念、及びステータス応答の使用例を踏まえた上で、従来のプリンタ装置における問題点を説明しよう。例えば、複数のドキュメント出力のジョブを処理する場合のステータス応答について考えてみる(図8)。

【0068】まずホストコンピュータ側での動きを説明する。今、ホストコンピュータなどが複数の出力ジョブをプリンタ装置から出力しようとしていたとする。複数のドキュメント出力を行なう場合、ホストコンピュータはそれを順番に一つ一つ処理していく。

【0069】まず一つめのドキュメント出力を開始する。ジョブの開始をプリンタに対し宣言し、プリンタ環境を初期化し、これから行なう出力のためのプリンタ環境設定などの前処理を行なう(1)。

【0070】次にホストコンピュータは、出力するドキュメントの内容を文書ファイルなどから読み込み、その内容から文字や図形やイメージなどをどのようにレイアウトすれば良いか判断し、その描画内容を該当するプリンタ言語記述形式に変換し、出力データとしてプリンタ装置に送り出す(2)。

【0071】全ページの出力データ転送が完了したら、ホストコンピュータは、プリンタ環境を元に戻し、ジョブの終了を宣言するなどの後処理を行なう(3)。これでホスト側において、最初のドキュメント出力、ジョブ1が終了した訳である。

【0072】ホストコンピュータは、続いて次のドキュメント出力、ジョブ2を処理開始する(4)。ジョブ2の処理もジョブ1と同様に行われる。さらにドキュメント出力が続く場合には、ジョブ3、ジョブ4と同様の処理を行なう。

【0073】次に、プリンタ側での処理について説明する。

【0074】ホスト側から(1)の前処理のデータを受け取るとその指令に従って、プリンタは環境の初期化、必要な環境の設定を行なう(1')。

【0075】この時、前述したステータス応答により、使用できる用紙サイズの情報や、プリンタの空きメモリ容量などの情報をホスト側へ連絡する。この情報によりホスト側では、最適なプリントが行なえるよう出力データを調節する。

【0076】このステータス応答は通常、ホストが要求した時にすぐプリンタ側から返されるので、ホスト側の処理(1)とプリンタ側の処理(1')はほぼ同期して

いる。

【0077】続いてホスト側から(2)の出力データを受け取る。プリンタ側では受け取ったこの出力データを解析し、対応する文字や図形やイメージの描画を行う。そして改ページの指令を受け取り、描画された用紙を排出する(2')。

【0078】この時、前述したステータス応答により、ページが排出される毎に何ページまで出力が完了したかの情報や、紙切れなどのプリンタの状態の情報をホスト側に伝達する。ホスト側ではこれを受け取り、プリンタ10がどのような状態であるか把握することが可能となる。

【0079】但し、このプリンタ側の出力処理は、文字や図形などの展開に時間がかかったり、プリンタエンジンの出力速度に制限があるため、どうしてもある程度の時間がかかる。このためホスト側では、(2)の出力データ作成処理及び、プリンタへの転送処理が完了していても、プリンタ側ではまだ(2')の描画展開処理及び、ページ出力処理を行なっている場合が多々ある。

【0080】ホスト側でのデータ作成、転送処理は1ページあたり通常数秒程度で済むのに対して、プリンタ側20での描画出力処理は、1ページあたり早くても数十秒、遅い場合には数分を要するものもある。

【0081】最近ではこの処理スピードの違いをカバーするために、プリンタ側に大量の受信バッファを持たせ、ホスト側から送られてきたデータをとにかくそのバッファに蓄積してしまい、ホスト側をできるだけ早くプリント処理から開放してやるようになっている。このためなおさら、ホスト側の処理は瞬時に終わってしまい、プリンタ側との処理スピード差は広がってきている。

【0082】このように描画出力処理の段階では、ホスト30側の処理(2)とプリンタ側の処理(2')は、大抵の場合同じタイミングでは進まない。プリンタ側の処理が遅れる傾向にある。

【0083】このプリンタ側の処理の遅れにより、ホスト側から指令された後処理(3)の命令は、すぐに処理されず、取り敢えず受信バッファに蓄積される。そしてプリンタ側で(2')の描画出力処理が全て完了した後で、実行される(3')。

【0084】同様に、ホスト側における(4)次のジョブの出力データも、取り敢えず受信バッファに蓄積され40るのが普通で、プリンタ側ですぐに処理される訳ではない。前のジョブが完了してから引き続いて処理されるようになっている。

【0085】従って、ホスト側では既に次のジョブの処理が行なわれているのに、プリンタ側では前のジョブの出力を行なっているような状況が頻繁に起こりうる。このようなホスト側とプリンタ側の処理のタイミングのズレが、ステータス応答において大きな問題を生じさせている。

【0086】例えば、ホスト側の処理が既にジョブ2の50

出力データ作成転送処理の段階に入っていた時、プリンタ側はジョブ1の描画出力処理の段階だったとする。

【0087】プリンタ側はページを排出する毎にホスト側に向けて、「現在3ページ目を排出完了した」といった情報をステータス応答として送出する。これは即ち「ジョブ1」の「3ページ目の排出」を意味するのだが、従来のステータス応答では「何ページ目であるか」という情報だけしか返していなかった。このためホスト側では、現在ホスト側で処理している「ジョブ2」の「3ページ目」などと誤解をしてしまう可能性がある。

【0088】ページの排出状態以外にも、例えばジョブ2で要求した登録フォントの種類の情報が、プリンタ側から応答された時には、既にホスト側ではジョブ3の処理をしていた、などといったように、あらゆるステータス応答において混乱を招く恐れを持っている。

【0089】近年では、一つのプリンタ装置を同時に複数のホストやプロセスやユーザが共用して使用する例が増えている。そのような環境では、このような混乱が頻繁に起こる。

【0090】また一対一の関係であっても、複数のジョブにおけるステータス応答要求コマンドと、それに対するステータス応答のタイミングが不確定であることは、複雑なステータス応答によるホストとプリンタ装置の情報交換を妨げ、ホストとプリンタの制御が一体となった高度なプリンタシステムの構築の障害となっている。

【0091】このように従来のプリンタ装置では、ステータス応答をホストに対して送出する時に、その応答がどのジョブにおけるものであるか、を識別する手段が無かったために、ホスト側での混乱を招き、ホスト側での正しいプリンタ状態把握を困難にするという問題点があった。

【0092】本発明は上記問題点に鑑みてなされたもので、ホストコンピュータ側で確実にその状態を正確に把握することができるプリンタ装置及びその制御方法を提供することを目的とする。

【0093】

【課題を解決する為の手段】この課題を解決するために、本発明のプリンタ装置は、データ元からのジョブ単位のデータに従ってプリント処理を行うプリンタ装置であって、前記ジョブ単位のデータを受信する受信手段と、該受信手段により受信したジョブ単位のデータそれぞれに固有の識別子を割り当てる割り当て手段と、前記受信手段により受信したジョブ単位のデータに対する処理の完了応答である情報を、前記割り当て手段で当該ジョブ単位のデータに対して割り当てられた識別子と共にデータ元に返信する情報返信手段とを備える。

【0094】また、本発明の制御方法は、データ元からのジョブ単位のデータに従ってプリント処理を行うプリンタ装置の制御方法であって、前記ジョブ単位のデータを受信する受信工程と、該受信工程により受信したジョ

ブ単位の前記データそれぞれに固有の識別子を割り当てる割り当て工程と、前記受信工程により受信したジョブ単位の前記データに対する処理の完了応答である情報を、前記割り当て手段で当該ジョブ単位の前記データに対して割り当てられた識別子と共にデータ元に返信する情報返信工程とを備える。

【0095】また、本発明のプリンタ装置は、データ元からのジョブ単位の前記データに従ってプリント処理を行うプリンタ装置であって、前記ジョブ単位の前記データ及び当該ジョブ単位の前記データ毎に割り当てられた固有の識別子を受信する受信手段と、前記受信手段により受信したジョブ単位の前記データに対する処理の完了応答である情報を、当該ジョブ単位の前記データに対して割り当てられた識別子と共にデータ元に返信する情報返信手段とを備える。

【0096】また、本発明の制御方法は、データ元からのジョブ単位の前記データに従ってプリント処理を行うプリンタ装置の制御方法であって、前記ジョブ単位の前記データ及び当該ジョブ単位の前記データ毎に割り当てられた固有の識別子を受信する受信工程と、前記受信工程により受信したジョブ単位の前記データに対する処理の完了応答である情報を、当該ジョブ単位の前記データに対して割り当てられた識別子と共にデータ元に返信する情報返信工程とを備える。

【0097】

【作用】上記構成により、プリンタ側でジョブ単位の前記データに固有の識別子を付し、各ジョブ単位の前記データに対する識別子と、各ジョブ単位の前記データに対する応答の識別子とをデータ源が認識して、たとえ、ステータス応答が送り返されるまでに時間が経過してしまっている、あるいは、複数のステータス応答が送り返される順番が、その応答を要求するジョブ単位の前記データの順番と異なってしまっている、どのジョブ単位の前記データに対する応答であるかを明確に提示することを可能とする。

【0098】また、上記構成により、データ源側でジョブ単位の前記データに固有の識別子を付し、各ジョブ単位の前記データに対する識別子と、各ジョブ単位の前記データに対する応答の識別子とをデータ源が認識して、たとえ、ステータス応答が送り返されるまでに時間が経過してしまっている、あるいは、複数のステータス応答が送り返される順番が、その応答を要求するジョブ単位の前記データの順番と異なってしまっている、どのジョブ単位の前記データに対する応答であるかを明確に提示することを可能とする。

【0099】

【実施例】

(実施例1) 以下、添付図面にしたがって、本発明の実施例を説明する。

【0100】図1は実施例のプリンタ装置の構成を示すブロック図である。

【0101】本実施例では、1実施例として、ページプリンタについて記述してある。

【0102】図において、1はホストコンピュータであり、文字、図形などの出力データ及び各種制御コマンドを発生する他、プリンタ装置の各種情報を知るためのステータス応答要求コマンドを発生し、出力装置の主要部をなすプリンタ装置2に入力する。

【0103】ホストコンピュータ1とプリンタ装置2の間は、双方向インターフェースにより接続されており、ホスト1からプリンタ2へのデータ転送の他、プリンタ装置2からホスト1へのデータの返信を可能としている。

【0104】プリンタ装置2において、3はデータ受信管理部であり、ホストコンピュータ1より入力される文字や図形などの出力データ、各種制御コマンド、ステータス応答要求コマンドを受け取り、これを管理する。

【0105】このデータ受信管理部3は、ホスト1からのデータを受け取り、一時的に蓄えておくための受信バッファを備えている。

【0106】4は、これらコマンドの内容を詳しく解析し、その命令に従って関連する制御を指示するプリンタ言語解析部である。ここでは、データ受信管理部3が保持している各コマンドを一つ一つ順番に受け取り、その内容をプリンタ言語の記述ルールに従って調査し、そのコマンドがどのような処理の実行を要求しているか判別する。

【0107】この判別の結果、そのコマンドがプリントジョブの開始、終了宣言などといった、ジョブ管理関係のコマンドであった場合には、ジョブ管理部5に対して該当するジョブ管理指令を発行する。

【0108】そのコマンドが文字や図形やイメージなどの描画を指令するものであった場合には、描画出力部6に対して該当する描画出力を指令する。

【0109】プリンタ装置の制御を指令するものであった場合には、プリンタ装置制御部7に対して該当する制御を指令する。

【0110】また、なんらかのステータス応答を要求するものであった場合には、ステータス応答処理部8に対して該当するステータス応答要求の指令を発行する。

【0111】5は、本発明によるところのジョブ管理部である。

【0112】ジョブ管理部5では、プリンタ言語解析部4からのジョブ管理指令に従って、ホスト1から送られてきたプリントデータをジョブ単位で処理する。本発明によるところのジョブ管理部5では、ホスト1よりプリントジョブのジョブ開始宣言を受け取ると、そのジョブに対して管理IDを自動的に割り振る。

【0113】この管理IDは、このジョブを他のジョブと区別するためのもので、例えば、ジョブ開始宣言を受け付けた順番に振られる番号であったり、或いは、何ら

かの独立した文字列であったりする。

【0114】但し、この管理IDは、受け付けたジョブ毎に全て異なるIDでなければならない。

【0115】そしてこのジョブ毎に割り振られた管理IDを、この受け取った出力データの各ジョブに対して、ジョブ管理ID情報として付加して管理する。このジョブ管理ID情報は、その対象となるジョブの処理が完全に完了するまで保持される。

【0116】さらにこれと同時に、ジョブ管理部5は、このジョブ管理ID情報を、双方向インターフェースを介して、即時にホストコンピュータ1へ返送する。

【0117】この時の返送は、ジョブを受け取った順番に、すぐに実行される。

【0118】6は、プリンタ出力を行なうための文字や図形やイメージなどの描画展開を行ない出力するための、描画出力部である。

【0119】ここでは、プリンタ言語解析部4からの描画出力指令に従って、該当する文字パターンを作成したり、図形の計算、描画を行ったり、イメージデータの展開を行ない、これら出力展開データをプリンタエンジン9に送出する。

【0120】プリンタエンジン9は、例えばレーザビームプリンタエンジンなどのような装置で、描画出力部6から受け取った出力展開データに従い、電子写真方式などの仕組みを用いて、実際のプリント出力処理を行なう。

【0121】7は、プリンタ装置制御部である。ここでは、プリンタ言語解析部4からのプリンタ装置制御指令に従い、例えば使用する用紙の選択や、プリンタ装置の初期化などといった、プリンタ装置の制御及び管理を行なう。

【0122】8は、本発明によるところのステータス応答処理部である。

【0123】ここでは、プリンタ言語解析部4から受け取ったステータス応答要求指令に従い、該当するプリンタ環境の内容を調査する。

【0124】調査する内容は、各種フォント情報や、描画機能の内容、データを受け付けられる状態かどうか、プリンタ装置のステータス、プリンタ出力の進行状態などなど、多岐に渡るものである。

【0125】このステータス応答処理部8では、これら各種プリンタ環境を調査し、その結果の内容に従って、文字列などの形式によりこの内容を表現するステータス応答情報を作成する。

【0126】この時、本発明によるプリンタ装置では、この応答要求命令が記述されていたジョブデータに対して、ジョブ管理部5が割り振った管理ID情報を、この応答情報に付加する。

【0127】そして、このステータス応答情報をデータ送信管理部10に伝達する。

【0128】データ送信管理部10では、ステータス応答処理部8から返されたこのステータス応答情報を受け取ると、送信バッファなどに一時的に蓄え、双方向インターフェースのタイミングを見はからって、この管理ID情報が付加された形式で、ステータス応答をホスト1に対して返信する。

【0129】図2は、図7の例に対応して、本発明によるステータス応答の一例を表すものである。

【0130】今、ホストコンピュータからプリンタに対して最初のジョブ1が送り出されたとする(1)。

【0131】プリンタ言語解析部4は受け取ったジョブデータを命令単位毎に解析し、ジョブの開始宣言を受け取ると、それをジョブ管理部5に伝達する。

【0132】本発明によるジョブ管理部5は、これに従ってジョブの管理を開始するとともに、このジョブに対してジョブ管理ID情報を割り振る(2)。

【0133】そして、ジョブ管理部5は、この割り振られた管理ID情報を付加した形で各ジョブを管理する。

【0134】例えば、最初のジョブ1に割り振られたジョブ管理IDが「37」、次のジョブ2が「38」、その次のジョブ3が「39」などといったように管理される。

【0135】ジョブ管理部5は、割り振ったジョブ管理IDを、即座にステータス応答処理部8から、ホストコンピュータ1に送出し、送られてきたジョブ1が、管理ID「37」であることを伝達する(3)。

【0136】これは例えば「STATUS JOB ID: 37 RECEIVED」といった文字列などにより伝達される。

【0137】前述の例で述べたように、ホスト1はプリンタの各種情報をステータス要求する(4)。

【0138】使用できる用紙サイズの種類や、用意されているフォントの内容、紙切れなどの障害が発生していないかなどといったプリンタの状態、現在出力中のジョブが何ページまで排出されたか、全ページ出終わったかどうか、などなどのあらゆる情報である。

【0139】これらに対するステータス応答は、対应用紙サイズの情報のようにすぐに送り返せるものもあれば、プリンタの状態のように刻々と変化する、その都度送り返さなくてはならないものや、ページの排出状況のように、プリンタエンジンが処理し終わるまで待ってから応答しなくてはならないものもある。

【0140】このため場合によっては、ステータス要求を受け取ってから、かなり時間が経ってからホストに送り返される場合がある。

【0141】そして、前述したようにホスト側での最初のジョブ1の処理が終わり、次のジョブ2の処理に移行していたりする場合がよく起こりえる。

【0142】そこで本発明によるステータス応答処理部8では、従来のステータス応答情報に、各ジョブに割り

振られた管理ID情報を付加した形式で、ホストコンピュータ1に返信を行なう(5)。

【0143】例えば今、ジョブ1の出力データを処理していて、3ページ目がプリント完了したとする。ページの排出状態のステータス要求がホストから指定されていた場合には、これに連動して、3ページ目排出完了の情報をホストに伝達しなければならない。

【0144】この時、本発明によるステータス応答処理部8は、このページ排出情報に、ジョブ1の管理ID「37」を付加し、例えば「STATUS JOB-ID: 37 PAGE: 3 FINISHED」といった文字列などによって情報を表し、ホストへ伝達する。

【0145】ホストコンピュータ側では、この従来のステータス応答に付加されたJOB管理IDとジョブ開始時に受け取っていた現在のJOB管理IDを比較することで、送り返されてきたステータス応答が、どのジョブデータ中で発生した応答であるか、容易に判断することが可能となる訳である。

【0146】例えばこの例の場合では、送られてきた「STATUS JOB-ID: 37 PAGE: 3 FINISHED」の情報から、このページ排出情報が、ジョブ管理ID「37」のデータのページ排出に関するものであることが判る。

【0147】ジョブ1が開始した時にプリンタ側から送り返されている、ジョブ1に割り振られたジョブ管理ID「37」と、このステータス応答に割り振られたIDは一致する。従って、このステータス応答は、ジョブ1のデータにより発生したものである。

【0148】これが判ることで、ホスト側では、既に処理が始まっている、ジョブ2、ジョブ3とは関係ない応答であると判断する訳である。

【0149】図3は、ホストコンピュータ1のうちの本発明に関連する部分の構成を示すブロック図である。

【0150】図において、1はホストコンピュータであり、2は前述したプリンタ装置である。

【0151】11は、プリント出力管理部で、ユーザなどからの指定に従い、ドキュメントなどのプリント出力全般を管理する。

【0152】ユーザなどがドキュメントなどのプリントを指示すると、プリント出力管理部11は、そのドキュメント出力を一つのプリントジョブとして管理する。そして指定されたドキュメントの内容を、例えばハードディスク12などに記録されているドキュメントファイルから受け取り、それをプリントデータ作成部13に渡す。

【0153】またプリントジョブを行う上で必要なプリンタ装置の情報などを得るために、ステータス要求指令をプリントデータ作成部13に送る。

【0154】プリントデータ作成部13は、プリント出力管理部11から受け取った、ステータス要求指令や、

実際のドキュメント内容を、プリンタ装置2が理解することの出来るプリンタ言語形式に変換してやり、これをホスト側データ送信部14に渡す。

【0155】ホスト側データ送信部14では受け取ったジョブデータを、プリンタ装置2に送出する。

【0156】プリンタ装置2では前述したように、これらジョブデータを解析し各種プリント処理を行う他、各種ステータス応答を返信してくる。

【0157】返信されたステータス応答は、ホスト側データ受信部15が受け取られ、これを応答データ解析部16に渡される。

【0158】応答データ解析部16では、受け取ったステータス応答を解析し、その内容をプリント出力管理部11に伝える。

【0159】ここでやり取りされるステータス応答は、以下のように処理される。

【0160】まずホスト1がジョブを送出するとプリンタ側は、そのジョブに対してプリンタ側2が割り振ったジョブ管理IDを返信してくる。

【0161】プリント出力管理部11はこれを受け取ると、現在処理しているジョブに対して割り振られた管理IDであると判断し、そのジョブに対応付けて記憶しておく。

【0162】次にプリンタ装置2は、ホスト1のステータス要求の指示に従って、適宜、ステータス応答を返信してくる。

【0163】前述したように、本発明による、この時のステータス応答には、同時にジョブ管理IDの情報が付加されている。

【0164】プリント出力管理部11では、各ステータス応答の内容と同時に、このジョブ管理IDを受け取る。そしてこのステータス応答に付加されていたジョブ管理IDと、ジョブ開始時に受け取っていたジョブ管理IDを比較する。

【0165】もし一致していれば、これは現在処理しているジョブ中に発生したステータス応答であり、利用することが出来る。もし一致していなければ、別のジョブ中に発生したステータス応答であり、無視することが出来る。並列処理が可能なホストや、複数のユーザやホストが接続されているような環境では、同時に複数のジョブをホスト側で処理することもありえる。このような場合にも、このジョブ管理IDにより、個々のジョブが要求したステータス応答だけを選び出して、プリンタの状態を知ることができる訳である。

【0166】このように、ジョブに割り振られた管理IDの情報を、従来のステータス応答に付加して、返信してやることにより、どのステータス応答が、どのジョブ中に発生したものであるかを推測することが可能となる。

【0167】なお、この実施例では、管理ID情報とし

て、通し番号形式のIDを例にし説明したが、これはどのようなIDであっても構わない。

【0168】要するに本発明の本質としては、プリンタ装置でジョブを管理しているIDが、ホストコンピュータに提示できるものであれば、どのようなものでも構わない訳である。

【0169】図4は本発明の動作を示すフローチャートであり、プリンタ言語解析部4が、コマンド単位にデータを処理し、各種ジョブ管理処理、各種描画処理、各種制御処理、及び、各種ステータス応答処理を行なうよう10すについて表現したものである。

【0170】なお、(1)～(10)は各ステップを示す。

【0171】プリンタ言語解析部4は、受信データ管理部3からデータを受け取る(1)。

【0172】そして、このコマンドが、どのような処理の実行を指令しているものであるかを判別する(2)。

【0173】各種ジョブ制御を指令している場合には、ジョブ管理部5に対してこの指令を伝達し、該当するジョブ管理処理を実行させる。そしてその指令が、ジョブ20開始宣言であるかを判定する(3)。

【0174】ジョブの開始を宣言するものであった場合には、そのジョブに対してのジョブ管理IDを決定し、これを管理する(4)。

【0175】また、この決定したジョブ管理IDを即座にホストコンピュータ1へステータス応答させる(5)。

【0176】ジョブ開始宣言以外のジョブ管理指令であった場合には、該当するジョブ管理処理を行う(6)。

【0177】各種描画制御を指令している場合には、描30画出力部6に対してこの指令を伝達し、該当する文字や図形やイメージなどの描画展開処理を実行させ、実際のプリンタ出力を行なう。各種プリンタ制御を指令するコマンドであった場合には、プリンタ装置制御部7に対し指令を伝達し、該当する各種プリンタ制御を実行する(7)。

【0178】各種ステータス応答を要求するコマンドであった場合には、ステータス応答処理部8に対して指令を伝達する。ステータス応答処理部8では、該当するプリンタ環境の状態を調査する(8)。

【0179】この調査は、大量の情報を探索したり、或は何らかのプリンタ状態の変化などといった、タイミングによって確定したりする場合もあり、即時に調査出来ない場合がある。このような場合には、ここで全ての処理を中断するのではなく、マルチタスク処理などの手法を用いることで、続く他のデータ処理を行いながら、この調査を平行して行い、調査が完了するのを待つ。

【0180】調査が完了すると、その結果を文字列などのステータス応答データ形式情報に変換する。

【0181】この時本発明によるプリンタ装置では、こ50

の応答要求をしたジョブに対する、ジョブ管理IDの情報をこの調査情報に付加し、ステータス応答データ形式の文字列などによる応答データを作成する(9)。

【0182】そして、このジョブ管理ID情報が付加された形式のステータス応答データは、データ送信管理部10に渡され、一時的に送信バッファなどに蓄えられ、双方向インターフェースのタイミングに従い、ホストコンピュータ1に返信される(10)。

【0183】プリンタ言語解析部4は、新たに次のコマンドをデータ受信管理部3より受け取り、解析し、以上に述べてきた処理を繰り返す。

【0184】以上の処理によって、ホストコンピュータは、プリントジョブにおいて、ステータス応答を要求するコマンドを発行し、これに対するステータス応答を受け取った場合に、前もって、そのプリントジョブに対して割り振られた、プリンタ内部でのジョブ管理IDを認識し、また、返信されるステータス応答の情報として、その応答要求のあったジョブの、ジョブ管理ID情報を参照することが出来るので、ステータス応答が返信されるまでに時間が経過してしまい、ホスト側のジョブ処理とのタイミングがズレてしまった場合にも、そのステータス応答が、どのジョブ中のステータス応答要求コマンドに対する返信であるかを容易に判別することが可能となり、プリンタ装置の状態を、ジョブ毎にホストコンピュータ側で確実に把握することが可能となる。

【0185】(実施例2)以下、添付図面にしたがって、本発明の他の実施例を説明する。

【0186】本実施例では、ジョブ管理IDをジョブ管理部3で自動的に割り振るのではなく、各ジョブデータと一緒にホストコンピュータ1から送られてきた、ホスト側で指定したジョブ管理IDを用いて、ジョブ管理する。

【0187】以下に図5を用いて処理の流れを説明する。

【0188】実施例2では、ホストコンピュータ1は、各ジョブのジョブ開始宣言(1)といっしょに、それぞれのジョブを識別するためのジョブ管理IDをプリンタに伝える(2)。

【0189】このジョブ管理IDは実施例1とは違い、40 ホスト側で決定したものである。

【0190】ジョブ管理部5は、ホストコンピュータ1から、ジョブ開始宣言とそのジョブに割り振られたジョブ管理IDを受け取ると、これをこれから送られてくるジョブデータに対応付けて管理する(3)。

【0191】ジョブ管理部5自体が、自分でジョブ管理IDを割り振ることはしない。

【0192】これ以外の処理は、前述実施例1と同様である。

【0193】即ち、ステータス応答に付加されるジョブ管理ID情報は、プリンタ装置2内で割り振ったもので

はなく、ホストコンピュータ1側で指定したものとなる。

【0194】図6は、この実施例2における動作を示すフローチャートである。

【0195】なお、(1)～(9)は各ステップを示す。

【0196】プリンタ言語解析部4は、受信データ管理部3からデータを受け取る(1)。

【0197】そして、このコマンドが、どのような処理の実行を指令しているものであるかを判別する(2)。

【0198】各種ジョブ制御を指令している場合には、ジョブ管理部5に対してこの指令を伝達し、該当するジョブ管理処理を実行させる。そしてその指令が、ジョブ管理IDの指定命令であるかを判定する(3)。

【0199】ジョブ管理IDを指定するものであった場合には、その指定された管理IDをそのジョブに対するジョブ管理IDであると判断し、以降このジョブをこの管理IDにより管理することとする(4)。

【0200】ジョブ管理ID指定以外のジョブ管理指令であった場合には、該当するジョブ管理処理を行う(5)。

【0201】各種描画制御を指令している場合には、描画出力部6に対してこの指令を伝達し、該当する文字や図形やイメージなどの描画展開処理を実行させ、実際のプリンタ出力を行なう。各種プリンタ制御を指令するコマンドであった場合には、プリンタ装置制御部7に対し指令を伝達し、該当する各種プリンタ制御を実行する(6)。

【0202】各種ステータス応答を要求するコマンドであった場合には、ステータス応答処理部8に対して指令を伝達する。ステータス応答処理部8では、該当するプリンタ環境の状態を調査する(7)。

【0203】この調査は、大量の情報を探索したり、或は何らかのプリンタ状態の変化などといった、タイミングによって確定したりする場合もあり、即時に調査出来ない場合がある。このような場合には、ここで全ての処理を中断するのではなく、マルチタスク処理などの手法を用いることで、続く他のデータ処理を行いながら、この調査を平行して行い、調査が完了するのを待つ。

【0204】調査が完了すると、その結果を文字列などのステータス応答データ形式情報に変換する。

【0205】この時本発明によるプリンタ装置では、この応答要求をしたジョブに対する、ジョブ管理IDの情報をこの調査情報に付加し、ステータス応答データ形式の文字列などによる応答データを作成する(8)。

【0206】そして、このジョブ管理ID情報が付加された形式のステータス応答データは、データ送信管理部10に渡され、一時的に送信バッファなどに蓄えられ、双方向インターフェースのタイミングに従い、ホストコンピュータ1に返信される(9)。

【0207】プリンタ言語解析部4は、新たに次のコマンドをデータ受信管理部3より受け取り、解析し、以上に述べてきた処理を繰り返す。

【0208】以上の処理によって、ホストコンピュータは、プリントジョブにおいて、ステータス応答を要求するコマンドを発行し、これに対するステータス応答を受け取った場合に、前もって、そのプリントジョブに対してホスト側で割り振っておいた、ジョブ管理IDを記憶しておき、また、返信されるステータス応答の情報として、その応答要求のあったジョブの、ジョブ管理ID情報を参照することが出来るので、ステータス応答が返信されるまでに時間が経過してしまい、ホスト側のジョブ処理とのタイミングがズレてしまった場合にも、そのステータス応答が、どのジョブ中のステータス応答要求コマンドに対する返信であるかを容易に判別することが可能となり、プリンタ装置の状態を、ジョブ毎にホストコンピュータ側で確実に把握することが可能となる。

【0209】

【発明の効果】以上説明したように、本発明によれば、ホストコンピュータは、ステータス応答を要求するコマンドを発行し、これに対するステータス応答を受け取った場合に、返信されるステータス応答の情報として、その応答の元となったステータス応答要求が、どのジョブ中に発行されたものであるかを識別するためのジョブ管理ID情報を参照することが出来るので、ステータス応答が返信されるまでに時間が経過してしまった場合にも、そのステータス応答が、どのジョブ中のステータス応答要求に対する返信であるかを容易に判別することが可能となり、プリンタ装置の状態をホストコンピュータ側で確実に把握することが可能となる。

【図面の簡単な説明】

【図1】本発明の一実施例としてのページプリンタ装置の構成を示すブロック図である。

【図2】同実施例により改善されたステータス応答処理を表す模式図である。

【図3】ホストコンピュータのうちの本発明に関連する部分の構成を示すブロック図である。

【図4】同実施例のプリンタ装置における、各種コマンド処理の動作を示すフローチャートである。

【図5】他の実施例により改善されたステータス応答処理を表す模式図である。

【図6】他の実施例における、各種コマンド処理の動作を示すフローチャートである。

【図7】一般的な出力ジョブの一例を表す模式図である。

【図8】従来のステータス応答処理の一例を表す模式図である。

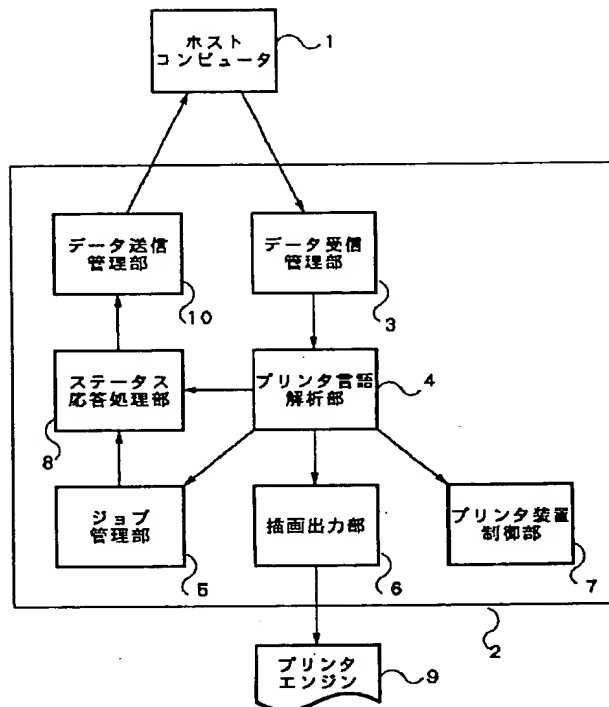
【符号の説明】

- 1 ホストコンピュータ
- 2 プリンタ装置

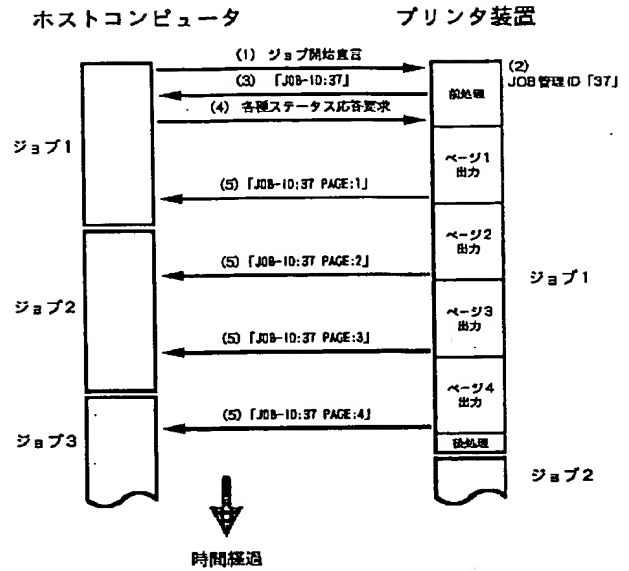
- 3 データ受信管理部
- 4 プリント言語解析部
- 5 ジョブ管理部
- 6 描画出力部
- 7 プリント装置制御部
- 8 ステータス応答処理部
- 9 プリントエンジン

- 10 データ送信管理部
- 11 プリント出力管理部
- 12 ドキュメントファイル
- 13 プリントデータ作成部
- 14 ホスト側データ送信部
- 15 ホスト側データ受信部
- 16 応答データ解析部

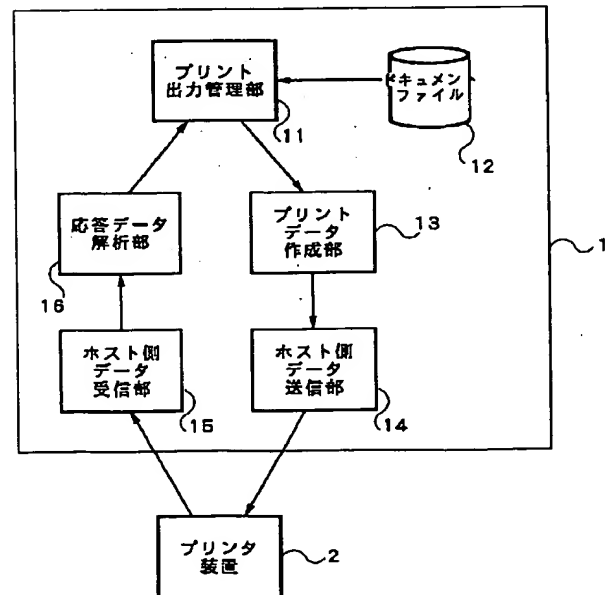
【図1】



【図2】



【図3】

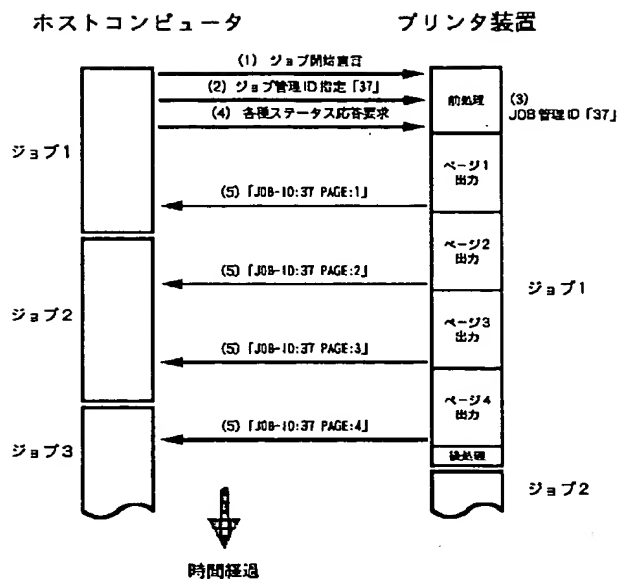


```

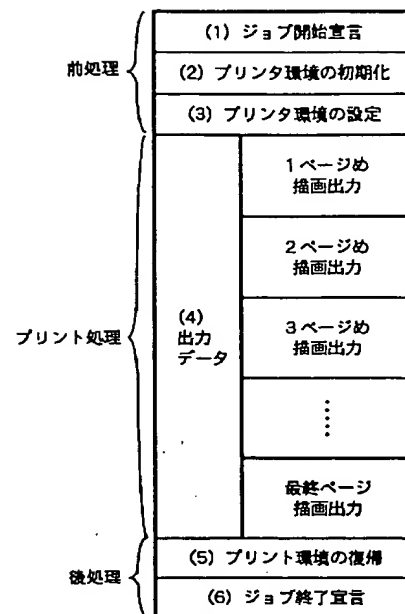
graph TD
    Start([開始]) --> J1(( ))
    J1 -- (1) --> S1[受信データ管理部  
からコマンド単位で  
データを受け取る]
    S1 --> D1{何の処理を  
指定しているか  
判別する}
    D1 -- (2) --> J2(( ))
    J2 -- ジョブ制御 --> D2{ジョブ開始宣言か?}
    D2 -- (3) YES --> S2[ジョブ開始処理を  
実行して、  
そのジョブに対して  
ジョブ管理ID  
を決める]
    D2 -- NO --> S3[その他の  
ジョブ管理処理  
を行なう]
    S2 -- (4) --> S4[決定した  
ジョブ管理IDを  
ホストに伝達する]
    S3 -- (6) --> J3(( ))
    J2 -- その他の  
プリンタ制御  
描画出力 --> S5[各種描画制御の  
指令に従い制御、  
描画処理を行なう]
    D1 -- (8) ステータス応答 --> S6[ステータス応答を  
行なうための  
プリンタ環境を調査]
    S6 -- (9) --> S7[ジョブ管理ID  
情報を付加して  
ステータス応答  
情報を作成]
    S7 -- (10) --> S8[ステータス応答  
をホストに返信]
    S4 --> J3
    S5 --> J3
    S8 --> J3
    J3 --> J1

```

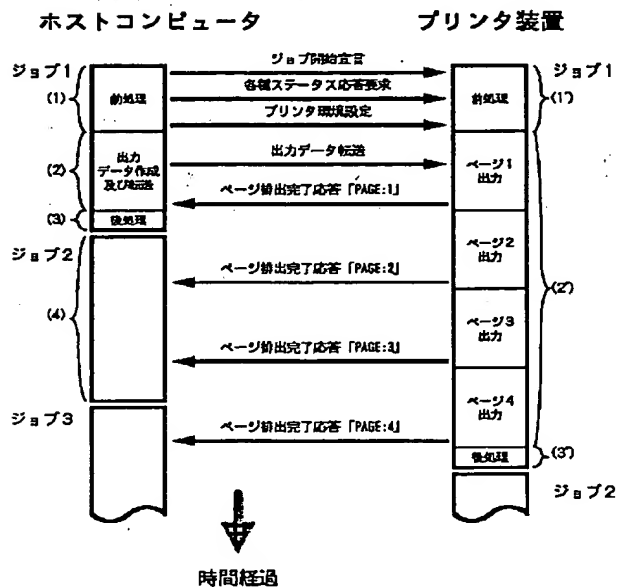
【図5】



ジョブデータ



【図8】



【図6】

